

IN THE CLAIMS

1. (Currently amended) A method of generating an application accessible by a user through one or more computer-based devices, the method comprising the steps of:

representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser; and

authoring the application using at least a portion of the interaction-based programming components;

wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices.

2. (Original) The method of claim 1, in a client/server arrangement wherein at least a portion of the application is to be downloaded from a server to at least one of the one or more computer-based devices, acting as a client, further comprising the step of including code in the application operative to provide a connection to the content/application logic resident at the server.

3. (Previously presented) The method of claim 2, wherein the code in the application operative to provide a connection to the content/application logic expresses at least one of one or more data models, attribute constraints and validation rules associated with the application.

4. (Original) The method of claim 1, wherein the one or more modality-specific renderings comprise a speech-based representation of portions of the application.

5. (Original) The method of claim 4, wherein the speech-based representation is based on VoiceXML.

6. (Original) The method of claim 1, wherein the one or more modality-specific renderings comprise a visual-based representation of portions of the application.

7. (Original) The method of claim 6, wherein the visual-based representation is based on at least one of HTML, CHTML and WML.

8. (Original) The method of claim 1, wherein the user interactions are declaratively represented by the interaction-based programming components.

9. (Original) The method of claim 1, wherein the user interactions are imperatively represented by the interaction-based programming components.

10. (Original) The method of claim 1, wherein the user interactions are declaratively and imperatively represented by the interaction-based programming components.

11. (Original) The method of claim 1, wherein the interaction-based programming components comprise basic elements associated with a dialog that may occur between the user and the one or more computer-based devices.

12. (Original) The method of claim 11, wherein the interaction-based programming components comprise complex elements, the complex elements being aggregations of two or more of the basic elements associated with the dialog that may occur between the user and the one or more computer-based devices.

13. (Original) The method of claim 1, wherein one of the interaction-based programming components represent conversational gestures.

14. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating informational messages to the user.

15. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating contextual help information.

16. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating actions to be taken upon successful completion of another gesture.

17. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating yes or no based questions.

18. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating dialogues where the user is expected to select from a set of choices.

19. (Original) The method of claim 18, wherein the select gesture comprises a subelement that represents the set of choices.

20. (Original) The method of claim 18, wherein the select gesture comprises a subelement that represents a test that the selection should pass.

21. (Original) The method of claim 20, wherein the select gesture comprises a subelement that represents an error message to be presented if the test fails.

22. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating rules for validating results of a given conversational gesture.

23. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating grammar processing rules.

24. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating dialogues that help the user navigate through portions of the application.

25. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating a request for at least one of user login and authentication information.

26. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating a request for constrained user input.

27. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating a request for unconstrained user input.

28. (Original) The method of claim 13, wherein the conversational gestures comprise a gesture for controlling submission of information.

29. (Previously presented) The method of claim 1, further comprising the step of defining logical input events and an association between the logical input events and physical input events that trigger the defined logical input events, such that the application may be authored using at least a portion of the definitions.

30. (Original) The method of claim 1, wherein the component by component transcoding is performed in accordance with XSL transformation rules.

31. (Original) The method of claim 1, wherein the component by component transcoding is performed in accordance with Java Bean.

32. (Original) The method of claim 1, wherein the component by component transcoding is performed in accordance with Java Server Pages.

33. (Canceled)

34. (Original) The method of claim 1, wherein representation by the interaction-based programming components supports a natural language understanding environment.

35. (Previously presented) The method of claim 1, further comprising the step of including code in the authored application for permitting cosmetic altering of a presentational feature associated with the one or more modality-specific renderings of the application on the one or more computer-based devices.

36. (Previously presented) The method of claim 1, further comprising the step of including code in the authored application for permitting changes to rules for transcoding on a component by component basis to generate the one or more modality-specific renderings of the application on the one or more computer-based devices.

37. (Original) The method of claim 1, wherein a definition of an underlying data model being populated is separated from a markup language defining the user interaction.

38. (Currently amended) The method of claim 1, A method of generating an application accessible by a user through one or more computer-based devices, the method comprising the steps of:

representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the

application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser; and
authoring the application using at least a portion of the interaction-based programming components;

wherein a node_id attribute is attached to each component and the attribute is mapped over to various outputs.

39. (Original) The method of claim 1, wherein an author is provided with a pass through mechanism to encapsulate modality-specific markup components.

40. (Original) The method of claim 1, wherein the components may be active in parallel.

41. (Original) The method of claim 1, wherein the representation and transcoding is extensible.

42. (Original) The method of claim 1, wherein a state of the application is encapsulated.

43. (Original) The method of claim 1, wherein the representation permits reference to dynamically generated data and supports callback mechanisms to the content/application logic.

44. (Currently amended) Apparatus for use in accessing an application in association with one or more computer-based devices, the apparatus comprising:

one or more processors operative to: (i) obtain the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and (ii) transcode the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance

with one or more modality-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser;

wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices.

45. (Original) The apparatus of claim 44, wherein the one or more processors are distributed over the one or more computer-based devices.

46. (Original) The apparatus of claim 44, in a client/server arrangement wherein at least a portion of the application is to be downloaded from a server to at least one of the one or more computer-based devices, acting as a client, further comprising the step of including code in the application operative to provide a connection to the content/application logic resident at the server.

47. (Previously presented) The apparatus of claim 46, wherein the code in the application operative to provide a connection to the content/application logic expresses at least one of one or more data models, attribute constraints and validation rules associated with the application.

48. (Original) The apparatus of claim 44, wherein the one or more modality-specific renderings comprise a speech-based representation of portions of the application.

49. (Original) The apparatus of claim 48, wherein the speech-based representation is based on VoiceXML.

50. (Original) The apparatus of claim 44, wherein the one or more modality-specific renderings comprise a visual-based representation of portions of the application.

51. (Original) The apparatus of claim 50, wherein the visual-based representation is based on at least one of HTML, CHTML and WML.

52. (Original) The apparatus of claim 44, wherein the user interactions are declaratively represented by the interaction-based programming components.

53. (Original) The apparatus of claim 44, wherein the user interactions are imperatively represented by the interaction-based programming components.

54. (Original) The apparatus of claim 44, wherein the user interactions are declaratively and imperatively represented by the interaction-based programming components.

55. (Original) The apparatus of claim 44, wherein the interaction-based programming components comprise basic elements associated with a dialog that may occur between the user and the one or more computer-based devices.

56. (Original) The apparatus of claim 55, wherein the interaction-based programming components comprise complex elements, the complex elements being aggregations of two or more of the basic elements associated with the dialog that may occur between the user and the one or more computer-based devices.

57. (Original) The apparatus of claim 44, wherein one of the interaction-based programming components represent conversational gestures.

58. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating informational messages to the user.

59. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating contextual help information.

60. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating actions to be taken upon successful completion of another gesture.

61. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating yes or no based questions.

62. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating dialogues where the user is expected to select from a set of choices.

63. (Original) The apparatus of claim 62, wherein the select gesture comprises a subelement that represents the set of choices.

64. (Original) The apparatus of claim 62, wherein the select gesture comprises a subelement that represents a test that the selection should pass.

65. (Original) The apparatus of claim 64, wherein the select gesture comprises a subelement that represents an error message to be presented if the test fails.

66. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating rules for validating results of a given conversational gesture.

67. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating grammar processing rules.

68. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating dialogues that help the user navigate through portions of the application.

69. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating a request for at least one of user login and authentication information.

70. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating a request for constrained user input.

71. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating a request for unconstrained user input.

72. (Original) The apparatus of claim 57, wherein the conversational gestures comprise a gesture for controlling submission of information.

73. (Previously presented) The apparatus of claim 44, further comprising the step of defining logical input events and an association between the logical input events and physical input events that trigger the defined logical input events, such that the application may be authored using at least a portion of the definitions.

74. (Original) The apparatus of claim 44, wherein the component by component transcoding is performed in accordance with XSL transformation rules.

75. (Original) The apparatus of claim 44, wherein the component by component transcoding is performed in accordance with Java Bean.

76. (Original) The apparatus of claim 44, wherein the component by component transcoding is performed in accordance with Java Server Pages.

77. (Canceled)

78. (Original) The apparatus of claim 44, wherein representation by the interaction-based programming components supports a natural language understanding environment.

79. (Previously presented) The apparatus of claim 44, further comprising the step of including code in the authored application for permitting cosmetic altering of a presentational feature associated with the one or more modality-specific renderings of the application on the one or more computer-based devices.

80. (Previously presented) The apparatus of claim 44, further comprising the step of including code in the authored application for permitting changes to rules for transcoding on a component by component basis to generate the one or more modality-specific renderings of the application on the one or more computer-based devices.

81. (Original) The apparatus of claim 44, wherein a definition of an underlying data model being populated is separated from a markup language defining the user interaction.

82. (Currently amended) ~~The apparatus of claim 44,~~ Apparatus for use in accessing an application in association with one or more computer-based devices, the apparatus comprising:
one or more processors operative to: (i) obtain the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and (ii) transcode the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser;

wherein a node_id attribute is attached to each component and the attribute is mapped over to various outputs.

83. (Original) The apparatus of claim 44, wherein an author is provided with a pass through mechanism to encapsulate modality-specific markup components.

84. (Original) The apparatus of claim 44, wherein the components may be active in parallel.

85. (Original) The apparatus of claim 44, wherein the representation and transcoding is extensible.

86. (Original) The apparatus of claim 44, wherein a state of the application is encapsulated.

87. (Original) The apparatus of claim 44, wherein the representation permits reference to dynamically generated data and supports callback mechanisms to the content/application logic.

88. (Currently amended) ~~The apparatus of claim 44,~~ Apparatus for use in accessing an application in association with one or more computer-based devices, the apparatus comprising:
one or more processors operative to: (i) obtain the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and (ii) transcode the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser;

wherein the one or more processors are distributed over the one or more computer-based devices and the application is synchronized across the one or more computer-based devices.

89. (Original) The apparatus of claim 44, wherein the representation of the application further permits cosmetization of the one or more modality-specific renderings via one or more modality-specific markup languages.

90. (Currently amended) A browser apparatus for use in providing access to an application by a user through one or more computer-based devices, comprising a machine readable storage medium containing computer executable code which when executed permits the implementation of the steps of:

obtaining the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and

transcoding the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser;

wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices.

91. (Currently amended) An article of manufacture for use in generating an application accessible by a user through one or more computer-based devices, comprising a machine readable storage medium containing computer executable code which when executed permits the implementation of the steps of:

representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and

further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser; and

authoring the application using at least a portion of the interaction-based programming components;

wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices.